University of Alberta

Segment-based Multistage Stereoscopic Depth Estimation

by

Leslie Jia

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2006

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-22293-5
Our file Notre référence
ISBN: 978-0-494-22293-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Stereoscopic analysis is widely used in machine vision applications. Local and global methods are two main branches of stereoscopic analysis. The global methods typically minimize a cost function over the entire scene. Although these methods provide high estimation accuracy, because of its high complexity, they are not suitable for real-time implementation. The local methods typically use window-correlation approaches, and the associated complexity is generally low. However, the estimation accuracy is sensitive to the selected window size. In this paper, we propose a multistage local method that operates on image segments instead of traditional rectangular windows. This new approach exploits the unique characteristics of image segments, and reduces occlusion through a feedback system. Experimental results show that it is very effective for natural images. In addition, it has a distinct advantage to preserve depth plane boundaries.

Acknowledgements

I would like to particularly express my gratitude to Dr. Mrinal Mandal for his guidance and help with my thesis research. I also thank Meghna Singh for her assistance and patience. My research would have been much harder without the research papers and test images listed on the Middlebury Stereo Vision Page. Thus, I thank all those who have contributed to that website.

This thesis is dedicated to my wife Ellen, and my parents Mei & Rong.

Contents

1 Introdution 1

1.1 Basic SDE Setup 3

1.2 The Correspondence Problem 4

1.3 Overview of the thesis 6

2 Related Work 7

2.1 Fundamental Constraints 7

2.2 Steps of a Generic Stereo Correspondence Algorithm 8

2.2.1 Matching cost 9

2.2.2 Cost Aggregation 12

2.2.3 Disparity Map Refinement 13

2.3 Common Obstacles 13

2.3.1 Featureless Regions and Repeating Patterns 13

2.3.2 Discontinuity 15

2.3.3 Occlusion 16

2.4 Global Optimization 17

2.5 Local methods 18

2.5.1 Shifting Blocks 20

2.5.2 Hierarchical Blocks 20

2.5.3 Image Segmentation 21

2.6 Review Summary 22

3 Proposed Multistage Algorithm 24

3.1 The Multistage Architecture 25

3.2 Image Segmentation 28

3.2.1 Image Feature Basics 28
3.2.2 Segment Correlation Methods 29
3.2.1 Unique Advantages of Segments 34
3.3 Implementing Segmentation and Matching Algorithms 35
3.3.1 Region Growing 36
3.3.2 Vector Quantization 37
3.3.3 Segment Matching 37
3.4 Left-Right Consistency Enforcement 38
3.5 System Integration 42
3.5.1 Image Preparation 42
3.5.2 Initial Disparity Range Estimaton 42
3.5.3 Segmentation and Matching 43
3.5.4 Post-processing 44
3.6 Summary 46

4 Performance Evaluation 47

4.1 Testbed Specifications 47
4.1.1 Test Images 48
4.1.2 Disparity Value Ranges 49
4.1.3 Performance Metrics 50
4.1.4 Test Procedures 50
4.2 Numerical Results 51
4.2.1 Advantages of multistage 51
4.2.2 Comparison with other methods 54
4.3 Visual Evaluation 56
4.4 Program Running Speed 60
4.5 Summary 61

5 Conclusions 62

5.1 Contributions 62
5.2 Publications 64
5.3 Future Work 64

5.3.1 Intelligent Segmentation 65
5.3.2 Surface Modeling 66
5.3.3 Parameter Seletion 66

6 Bibliography 67

List of Tables

3.1	Matching Scenarios: Rectangular Windows vs. Segmentation	33
3.2	Window Matching vs. Segment Matching	38
4.1	Disparity Ranges for Test Images	49
4.2	Performance Improvements Using Multiple Stages	53
4.3	Performance and area of determined pixels for Tsukuba	53
4.4	Performance and area of determined pixels for all test images	54
4.5	Performance Comparison with Existing Techniques	55
4.6	Program Running Speed: Time Spent (s) for Each Step	61

List of Figures

1.1	Different branches of range detection	2
1.2	Basic layout of a stereoscopic system	3
1.3	A typical left-right image pair and its ground truth depth map	4
2.1	Building blocks of the correspondence process	8
2.2	Block correlation search	9
2.3	Mismatches that cause aliases	14
2.4	Depth discontinuities within a single window	15
2.5	Occluded regions do not have corresponding counterparts	16
2.6	Shifting blocks are used to find image features	20
2.7	Hierarchical blocks are used to reflect depth discontinuities	21
2.8	Segment boundaries naturally follow depth discontinuities	22
3.1	General structure of a multistage approach	26
3.2	A flow chart of disparity calculation steps	27
3.3	Several aspects of image features	28
3.4	Scenarios of cost curves	30
3.5	Asymmetric cost curves	39
3.6	Feedback loop for left-right consistency enforcement	40
3.7	Examples of artefacts	45
4.1	Left images of the test image pairs	48
4.2	Mask images for evaluation purposes	52
4.3	Tsukuba output depth maps at different stages	56
4.4	Tsukuba depth maps generated by different SDE techniques	57
4.5	Sawtooth output depth maps at different stages	58
4.6	Venus output depth maps at different stages	59
4.7	Map output depth maps at different stages	60

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
ALL	All Image Pixels Except Occluded Pixels
DISC	Discontinuity Region Pixels
DM	Depth Map
FL	Featureless Region Pixels
PWMP	Percentage of Wrongly Matched Pixels
SAD	Sum of Absolute Difference
SDE	Stereoscopic Depth Estimation
SSD	Sum of Squared Difference

Chapter 1

Introduction

For centuries, stereoscopy has been an interesting topic for researchers. As a prefix, "stereo" means three-dimensional; the word "scope" refers to the range of one's perceptions. Eyes, the human visual system, are a perfect example of stereoscopy at work. Combining the left eye's view with the right eye's view, a person can easily obtain the range information of the views and thus form a 3-D perception.

With the invention of computers and advancement in robotic technology, there arises the need for sophisticated machine vision methods. Indeed, a computer performs better than a human where the task is too repetitive, too hazardous or too precision-oriented. Real world applications that require computer assisted range detection include, but are not limited to: satellite terrain mapping, robot intelligence and image rendering.

Fig. 1.1 shows the most common approaches of range detection. The active techniques would first project pulses of electromagnetic or sound waves, and then analyse the reflected pulses. The passive techniques receives optical signal without

transmitting. Generally speaking, active techniques are more accurate. Yet, the passive techniques are more versatile; they may be needed when active projecting is not possible or not allowed. For example, laser projection may not function well on a satellite due to the extraordinary distance between the satellite and the earth surface. When there are humans involved, laser can also be hazardous to human eyes. As a result, passive ranging techniques are still very popular.

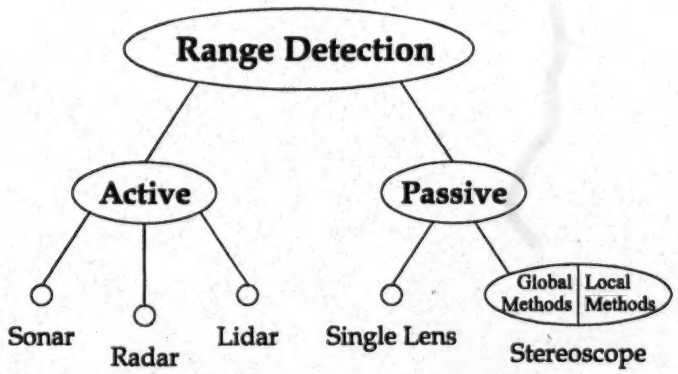


Fig. 1.1 Different branches of range detection.

There are several sub-branches within passive range detection techniques. With a single lens camera, range information can be obtained by analyzing the optical flow from a sequence of images. Stereoscopic range detection is far more common.

For most ranging applications, the term "depth" can usually be used interchangeably with "range". In fact, "depth" is the norm for image-based techniques. Therefore, we will use the phrase "stereoscopic depth estimation" (SDE) throughout this thesis. The depth information for an image is called a depth map.

By convention, common SDE applications have a left-right two lens setup, similar to human eyes. The left depth maps are regarded as the outputs of the SDE algorithm and are compared across different SDE algorithms.

1.1 Basic SDE Setup

In general, a stereoscopic application can have two or more cameras positioned at different locations. For simplicity, this thesis considers only the two-camera case, where they are positioned in parallel. As shown in Fig. 1.2, a mushroom is projected onto a pair of 2-D images via a pair of optical lenses.

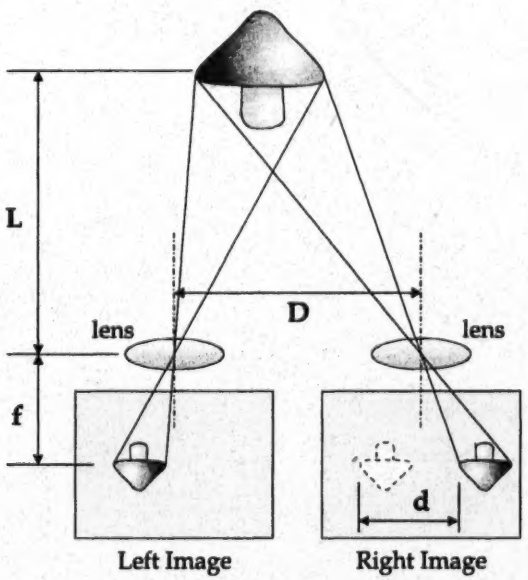


Fig. 1.2 Basic layout of a stereoscopic system

Here, L represents the distance between the object and the center of the two lenses; f is the focal length of the two identical lenses, which are separated by D; d,

the displacement between the object's projections on the two images, is known as disparity. We assume that the central axes of the two identical lenses are parallel and there is no skewing effect on the image planes.

It can be shown easily that the parameters L, D, f and d in Fig. 1.2 satisfy the following equation:

$$\frac{L}{D} = \frac{f}{d} \tag{1.1}$$

In a typical stereoscope setup, the focal length f and the lens displacement D are constants. Therefore, Eq. 1.1 can be expressed as $L = C / d$, where C is a constant. The task of determining L, essentially, becomes the estimation for the inverse of d. As a result, stereoscopic analysis can also be called "disparity analysis."

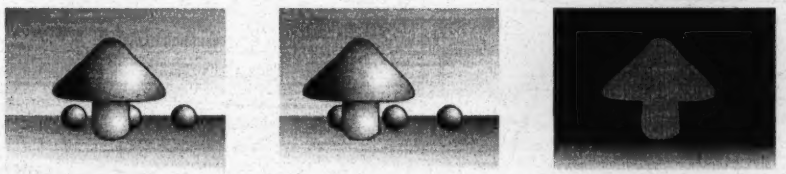
Note that the human visual system is slightly different from the above model. The "lenses" of human eyes will tilt a little bit such that both axes meet at the object. For objects far enough, i.e. $L >> D$, the difference between the two geometric models becomes diminishingly small. A completely revised model is needed, when the angle between the lenses is significant. However, this is too complex and is beyond the scope of our research.

1.2 The Correspondence Problem

As previously discussed, the physical world can be modeled as a 3-D composition of objects for most practical purposes. A natural image can be viewed as a 2-D planar optical projection of that 3-D scene. The projecting process is always unique and loss of information is unavoidable. When we observe this process closely, it is trivial that much of the information loss occurs in the dimension that is perpendicular to the

image plane. By convention, we call that dimension either the z axis or the d axis, while the 2-D image exists in the x-y space.

Fortunately, all information in that dimension is not lost. With two or more images, which are captured at different positions and yet focus on the same objects, we can attempt to partially recover the original 3-D composition. These images generally differ slightly, besides their strong correlations. Spatial diversities among the images are what we exploit in this reverse procedure.



Left Image Right Image Left Depth Map

Fig. 1.3 A typical left-right image pair and its ground truth depth map.

Fig. 1.3 shows a left-right image pair depicting a mushroom and three spheres and the grey level depth map corresponding to the left image. The depth map shown is simply the disparity values multiplied by a scalar. A darker pixel corresponds to a smaller disparity value, indicating a background (far) pixel. A brighter pixel corresponds to a larger disparity value, indicating a foreground (near) pixel.

Humans can easily establish a correspondence between the mushroom in the left image and the mushroom in the right image. It is this correspondence that leads to the determining of disparity values. However, what is trivial for humans can be quite complex for a computer to handle. A computer perceives nothing but a 2-D array of intensity levels. It is unrealistic to expect a computer to recognize objects

and make a reliable correspondence decision. Thus, SDE is also known as the correspondence problem. Without object recognition capabilities, an SDE algorithm relies on establishing correspondence for a single pixel at a time. Sophisticated aggregation methods are used to eliminate spurious disparity values and thus producing an acceptable depth map.

1.3 Overview of the Thesis

This thesis is about range detection by stereoscopic means, proposing a novel multistage architecture for the stereoscopic depth estimation process. The first stage determines the disparity values for large and uniform image segments using well established shift-matching algorithms. Each subsequent stage deals with smaller segments. Eventually, the entire depth map is to be completed.

This approach is quite similar to some image compression techniques, since low frequency signals are processed first. Each additional stage processes higher frequency image details, until the improvement becomes diminishingly small. In addition, segment based matching cost aggregation and left-right consistency enforcement methods are used to improve the performance further.

In chapter 2, relevant research works are reviewed. Multiple global optimization and local estimation techniques are explained; common scenarios are discussed. In chapter 3, the proposed algorithms and the associated implementation are covered. The proposed method is implemented and tested in Java. The results indeed demonstrate substantial improvements from earlier single stage algorithms of a similar approach. In chapter 4, the testing metrics and results are presented. Chapter 5 concludes this thesis and points to the potentials for future research.

Chapter 2

Related Work

For centuries, people have known that a 3-D scene composition can be deduced from both eyes' views. The mapping of a 3-D scene onto a pair of images is unique, but the reverse is not so. Some of the range information is lost forever and cannot be recovered. In other words, theoretically, a left-right pair of stereoscopic images may not contain sufficient information to determine a unique and correct depth map. Such difficulties, compounded with the stereoscopic correspondence problem, pose a great challenge to researchers.

2.1 Fundamental Constraints

The first rigorous examination of this subject, however, is D. Marr and T. Poggio's 1977 milestone paper [MP77]. The authors introduced the two fundamental constraints in stereoscopic depth estimation based on the underlying geometry. The first constraint is known as the uniqueness constraint, which states that any point in one of the images has either one or zero corresponding point in the other image. The sec-

ond one is called the smoothness constraint, which states that the 3-D scene is composed of piecewise smooth surfaces. Adjacent points are likely to have similar disparities.

Note that the uniqueness and smoothness constraints counter balance each other. Good depth estimations can only be achieved when the two constraints are balanced. This forms the theoretical basis for most of today's stereoscopic algorithms. Local methods and global methods are the two major branches in stereoscopic analysis. Their main differences lie between their ways of modeling the two constraints. Subsequent research efforts are mainly focused on either new modeling techniques or new computation methods for a known model.

2.2 Steps of a Generic Stereo Correspondence Algorithm

Numerous stereo correspondence algorithms have been proposed in the literature. Yet, most of them share the same basic steps. Fig. 2.1 depicts the flow diagram of a typical stereoscopic correspondence setup [SS02]

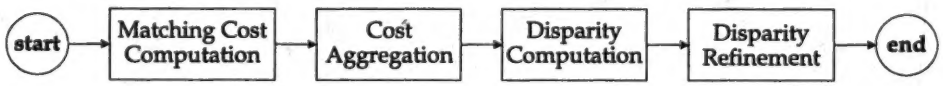


Fig. 2.1 Building blocks of the correspondence process.

As shown, a typical stereo algorithm is characterized by its matching cost definition, its cost aggregation, and its computation and refinement choices. It is definitely a generalization, since different methods put their emphasis on different com-

ponents. However, this should be a reasonable common ground to begin with. The two fundamental constraints play a significant role here. The matching cost computation process is a means to address the uniqueness constraint, while the smoothness constraint is enforced by cost aggregation. Every step is discussed in detail in this section.

2.2.1 Matching Cost

In order to be familiar with the terminologies and concepts used frequently in a stereo analysis, let us first look at the block matching algorithm. This method is probably the most basic in detecting the desirable d, and thus the inverse of d. Refer to Fig. 2.2. The left image is divided into multiple rectangular blocks. All pixels of the same block are assumed to have one uniform disparity.

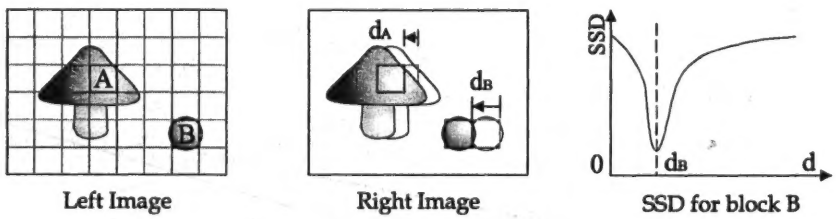


Fig. 2.2 Block correlation search

The task is to find the matching counterpart for every block in the right image. According to the previous geometric analysis, if the image pair is well calibrated, the vertical direction is totally irrelevant here. We can find the matching blocks by shifting the blocks from right to left in right image. Similarly, if we choose to divide the right image into blocks, we need to shift them from left to right in the left image.

The search typically starts from the reference position, i.e. $d = 0$, and gradually shifts horizontally, one pixel at a time. The search is over when a preset threshold, which represents the maximum disparity allowed, is reached. For each position, a similarity check is performed. Eventually the most similar block, based on correlation measures, is chosen. After proper scaling, the inverse of its disparity is then said to be the distance of that block. A completed depth map is obtained, after such a procedure is performed upon every pixel.

A function is always defined to quantify the degree of existing correlation. It is also known as the matching cost function; it serves as an essential building block of the correspondence process, as indicated previously in Fig. 2.1. By convention, a lower cost indicates higher correlation and a higher cost indicates lower correlation.

Mathematically, an image region S contains one or more pixels; each pixel has a scalar or vector intensity value. Let us consider the scalar case first, for simplicity. Note that S can be as small as one pixel, or as large as the entire image. In the case of block matching, S is simply a rectangular block of pixels. Designate $I_R(x, y)$ as the intensity function for the right image and $I_L(x, y)$ for the left image. Then, there exist the following frequently used matching cost functions [SS02].

Sum of Absolute Difference (SAD) The absolute value of the intensity difference of each corresponding pixel pair is obtained, and then the cost is summed within the entire block. It is equivalent to the Mean Absolute Difference (MAD), only differing by a scalar size factor.

$$\sum_{pixel \in S} \left| I_R(x-d, y) - I_L(x, y) \right| \tag{2-1}$$

Sum of Squared Difference (SSD) The square value of the intensity difference of each corresponding pixel pair is obtained, and then the cost is summed within the entire block. It is equivalent to the Mean Squared Error (MSE), only differing by a scalar size factor.

$$\sum_{pixel \in S} \left| I_R(x-d,y) - I_L(x,y) \right|^2 \tag{2-2}$$

Binary Cost The absolute value of the intensity difference of each corresponding pixel pair is obtained and compared to a pre-defined cost threshold T_C. The cost for that particular pair is either 1 or -1, depending on the comparison result. Eventually, the cost is summed within the entire region.

$$\sum_{pixel \in S} \operatorname{sgn}\left(\left| I_R(x-d,y) - I_L(x,y) \right| - T_C \right) \tag{2-3}$$

Gradient-Based Cost The cost function operates on the image gradient map; the function itself can use any of the mentioned cost. This approach can reduce the inaccuracy introduced by camera gain or bias. However, its use is not mandatory, as a pre-processing stage can perform an equalization procedure on the captured images.

$$\sum_{pixel \in S} f\left(I_R'(x-d,y), I_L'(x,y) \right) \tag{2-4}$$

All of the mentioned matching costs assume that d is an integer, which results in a horizontal translation in integer pixel width. However, due to image sampling, while the ground-truth image is a continuous and smooth map, the resulting disparity function is actually a discrete map. S. Birchfield and C. Tomasi proposed a way of eliminating such a discrete effect by linearly interpolating both R and P [BT98]. Thus, d does not have to be an integer anymore; sub-pixel precision can be achieved at the cost of more computational complexity.

2.2.2 Cost Aggregation

As mentioned above, a single pixel or a small homogenous region S is likely to encounter the aliasing problem. Spurious results are thus produced. A pixel, by definition, is really a single point with one intensity/color and has no feature inside. Thus, the correspondence problem is at its worst in the case of single pixel analysis—there may be multiple match aliases along the scan line.

Now, what if we consider a relatively large region? As the subject region grows larger, it is more likely that the region contains some sort of features. In the extreme case, when S represents the entire image, the region should have the most possible features. Therefore, we have to enforce the smoothness constraint, either in the form of summation over a larger region or in the form of minimizing an energy equation. This process of relating one pixel's disparity with other pixels' is called aggregation.

In the case of block matching, S is a rectangular block of pixels. The width and height of each block are design issue to be considered. This process of aggregating the costs of multiple pixels forms the second building block of the estimation process.

The fundamental idea behind aggregation is that the disparity value of a single pixel can be quite unpredictable due to aliases. However, according to the continuous constraint proposed by D. Marr and T. Poggio, the disparity map can be viewed as a piecewise-continuous surface. In other words, adjacent pixels are likely to have the same disparity. Thus, the aggregation process, which is essentially a low pass filter, can be employed to reduce spurious behaviours.

2.2.3 Disparity Map Refinement

The disparity value for each pixel is subject to error, since mismatches are not uncommon for occluded or featureless regions. Refinement techniques are available to eliminate obvious artefacts and smooth out uneven scan lines. Various low-pass filtering methods are used to reduce artefacts. 3-D surface reconstruction is very useful in refining the disparity map to sub-pixel levels, i.e. the disparity value of a pixel does not have to be rounded off to the nearest integer, as it can take on fractional values through surface fitting.

2.3 Common Obstacles

The shift-matching process is not always straight forward. Problems arise when a region is wrongly matched or has no match at all. In this section, we discuss several common obstacles involved in the matching process.

2.3.1 Featureless Regions and Repeating Patterns

Some typical matching costs are defined in the previous segment. Let us first check some extreme cases. When the subject region S contains exactly one single pixel, the summation is not needed. We may want to ask: why not just obtain the disparity of one pixel at a time, why not try to match 1 by 1 block pairs?

To answer that question, we need to consider the concept of image features. A single pixel, or a homogenous region, contains no features. In other words, within the subject area, the intensity is almost constant and its first partial derivative function is close to zero.

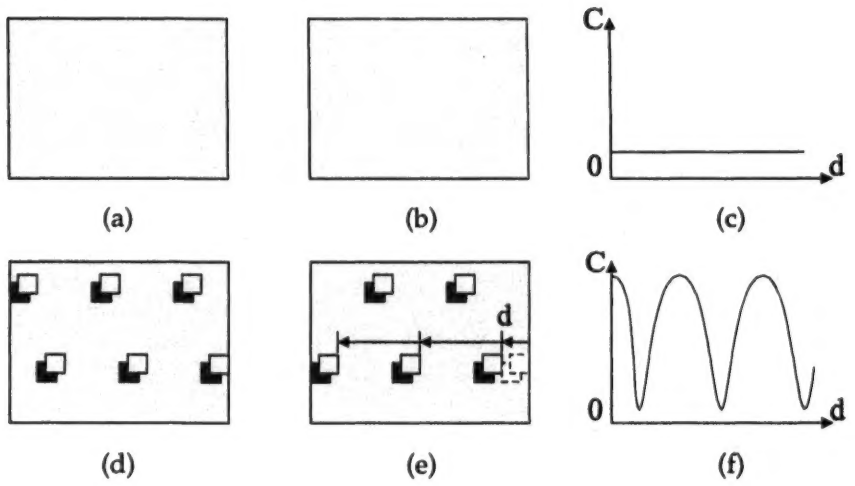


Fig. 2.3 Mismatches that cause aliases: (a) featureless left image, (b) featureless right image, (c) featureless cost curve, (d) repeating pattern left image, (e) repeating pattern right image, (f) repeating pattern cost curve.

Examples of featureless and repeating pattern image pairs are shown in Fig. 2.3. The top row shows a pair of featureless images (a, b). The featureless regions could be part of a blue sky, a blank wall, or a dark cave. When a region or the entire image is featureless, theoretically, there is no way to properly find its disparity. The matching cost curve, shown in (c), is likely to be a straight line that demonstrates high correlation (low cost) for all possible d values. In this case, even a human being would not be able to estimate the distance. The standard procedure in dealing with such regions is to perform a low pass filtering to fill the gaps.

The second row in Fig. 2.3 shows an image pair that contains a repeating pattern and its cost curve. In general, such image pairs produce matching cost curves that are periodic in nature. The real match may not have the lowest matching cost. In

that case, the computer algorithm would choose the alias with the lowest cost and thus resulting in a wrong match. The featureless situation can be viewed as a special case of the repeating pattern situation, where aliases exit for every increment of disparity d.

2.3.2 Discontinuity

A significant limitation of block matching is that we assume there is only one disparity value within each block. Refer to Fig. 2.4. There are three depth surfaces: a bright foreground sphere, a dark sphere and a blank background. For simplicity reasons, let us assume uniform disparity within the boundary of the same object. There are several blocks that contain parts of more than one object. Significant errors are unavoidable no matter how the matching process goes.

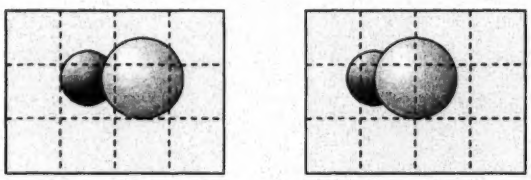


Fig. 2.4 Depth discontinuities within a single window.

As a rectangular block decreases in size, it would be less likely to contain more than one object. In the extreme case, for a block size of 1 by 1, each block is essentially a pixel and it is guaranteed to have one disparity only. However, as block size decreases, the likelihood of featureless blocks increases. Aliases can occur; a more sophisticated approach is needed.

2.3.3 Occlusion

As shown in Fig. 2.5, some regions in the left image do not have a match in the right image, because they are occluded by foreground objects. Though not shown in the diagram, certain regions in the right image could be occluded in the left image as well. This phenomenon is called occlusion. It is always difficult to find the disparity of the occluded regions, since it is not possible to know what exactly happens to that region in one of the images where it is invisible.

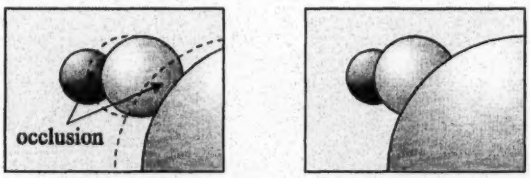


Fig. 2.5 Occluded regions do not have corresponding counterparts.

For most situations, that disparity is likely to be the same as the adjacent background object. However, this assumption is not always true. For instance, if the background object does not have uniform disparity within its boundary, then there is no solution.

There is a special case of the occlusion problem. As we can see from Fig. 2.5, part of the largest sphere, while visible in the right image, is invisible in the left image. That area is not occluded by another object; the occlusion is due to the fact that the area is off camera.

2.4 Global Optimization

D. Marr and T. Poggio proposed to use the uniqueness and continuity criteria to partially solve the correspondence problem [MP79]. Their approach only considered high-confidence features (mainly sharp edges), because even corresponding region can have highly different intensity levels due to shading distortion. After a sparse disparity map is obtained, the "empty" regions' disparities can be estimated using continuity methods, assuming that real world objects are cohesive.

Later research activities are divided into two main approaches. The first one is focused on the global optimization of the image disparity functions, while the second approach is focused on processing local image features. Their main differences lie between their ways of modeling the two constraints. According to D. Scharstein and R. Szeliski [SS02], the global approaches typically define an energy function that depicts the balance of the uniqueness and smoothness constraints.

Most of the work is performed during the disparity computation step, while aggregation is less important. A typical problem formulation is defined as the minimization of the energy function:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \tag{2-5}$$

The d value that makes $E(d)$ be at a minimum, or at least a local minimum, is regarded as a good disparity candidate for that particular pixel. The individual terms are defined as:

$$E_{data}(d) = \sum_{(x,y)} C(x, y, d(x, y)) \tag{2-6}$$

$$E_{smooth}(d) = \sum_{(x,y)} \rho(d(x,y) - d(x+1,y)) + \rho(d(x,y) - d(x,y+1)) \quad (2\text{-}7)$$

Optimal results are obtained when the energy function is minimized. Although the problem formulation of global approaches is simple and elegant, there is no trivial solution; the computational complexity is very high. Thus, research works in this area focus on improving computational techniques such as dynamic programming.

2.5 Local Methods

Although there are many obstacles, the block matching approach has its merits. First, it is easy to implement and fast running. Second, it provides reasonably good performance for images with rich features and relatively uniform in disparity. A number of methods have been employed to improve the performance.

Thus, the second approach focused on local image information and this approach itself can be divided into two branches. The first branch is feature-based. The intensity levels of the pixels are not trusted, since complex lighting and shading effects can occur. However, features such as sharp edges are thought to yield more consistent results. Researchers focused on obtaining better feature definition methods [C86, HS88]. The resulting disparity map of this approach is unavoidably sparse. That is the cost of high-confidence feature matching. Various methods have been investigated to "fill" into the holes of the sparse disparity map. They all assume the basic idea that the disparity map should be generally continuous and smooth [DA89].

The second branch is area-based. This approach eliminated the need to "fill" the disparity map; it is also referred to as the dense-disparity approach. Windows that contain features from the left image are selected and compared to windows

from the right image, or vice versa. There are two key issues to consider. First, window size/shape selection. The most straight forward approach is to have a fixed sized rectangular window and fixed location windows. However, this over-simplification may result in either featureless windows or a window that contains objects with different actual depths. In order to reduce the problem, different window types have been proposed [BI99, GLY95]. They examined a shifting window concept. The basic idea is that, when a window is thought to contain no good features, it is shifted horizontally or vertically until it contains a good feature.

When both M and N are as large as possible, the block is the entire image. Then, some questions arise. Why choose one block definition over another? Why choose a rectangular region at all? Other researchers proposed windows with adaptive size; it is also called the hierarchical approach. T. Kanade and M. Okutomi proposed to vary the window size based on a statistical model of intensity and disparity [KO94]. The end result is such that larger windows are used for featureless regions and smaller windows are used for regions with high-contrast boundaries or patterns.

However, all of the mentioned window-matching algorithms have a fundamental flaw. They only work for regular shaped objects such as a sphere; they fail most of the times in determining the depth of a rod-type object. In that case, no matter what the rectangular window size is, there will always be multiple significant objects of different depths within one window. Random-shaped window matching is the new trend, probably due to the increased ease to perform these experiments in recent years. The whole idea is based on the assumption that a uniform-colored region will likely have a continuous depth. While that is not entirely true, it is definitely superior to a rectangular window.

H. Tao et al proposed to roughly estimate the depth map first and then perform an image-segmentation [TSK01]. Each segment (window) is then fine-tuned via a global optimization algorithm to minimize the effect of occlusion. Eventually, each segment is modeled as either a frontal-plane—a surface parallel to the x-y plane—or a 3-dimesional flat surface or a 3-dimensional curved surface. Smoothness and color similarity constraints are enforced to achieve a smooth depth map.

2.5.1 Shifting Blocks

In order to solve, or at least partially solve, the problems caused by featureless blocks, researchers have proposed the use of shifting blocks. The idea is that if a block is featureless, then it is shifted horizontally and/or vertically until it contains substantial features, as shown in Fig 2.6. After the block in the bottom-right corner is shifted towards the sphere boundary, more image features are included in the block.

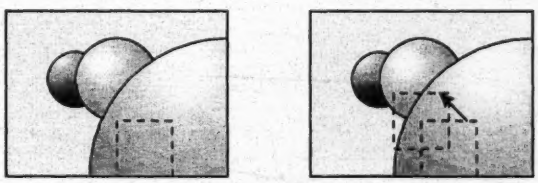


Fig. 2.6 Shifting blocks are used to find image features.

2.5.2 Hierarchical Blocks

As shown in section 2.3.2, some blocks contain multiple significant objects. In order to make the disparity uniform within a single block, researchers make use of a hierarchical block approach.

As shown in Fig. 2.7. The image is initially divided into large blocks. The algorithm then checks whether a block should have uniform disparity. If not, that block is divided into smaller blocks. The recursion continues until either the block is uniform in disparity, or when a preset lower limit of window size is reached.

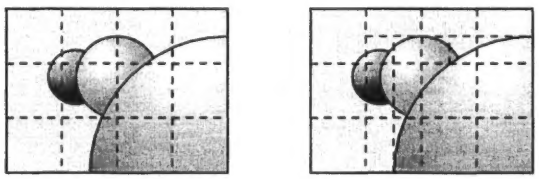


Fig. 2.7 Hierarchical blocks are used to reflect depth discontinuities.

The hierarchical block approach can also be used in combination with the shifting block approach. The result is a more precise and less blocky depth map.

2.5.3 Image Segmentation

Although the performance of the block matching algorithm can be improved substantially by employing rectangular blocks of various sizes and at arbitrary position, obstacles presented in section 2.3 are still significant. An intrinsically better approach is to use regions. Image segmentation techniques can be used to divide the whole image into regions of similar color, intensity and pattern. Each region is assumed to have a uniform disparity within its boundaries.

We assume that natural objects are uniform in terms of both color/intensity and depth. That assumption holds true for most cases. As illustrated in Fig. 2.8, the

advantage of this approach is that the boundaries are well preserved; the resulting disparity map is sharper and less blocky.

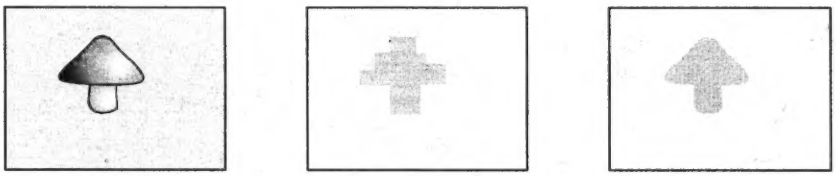


Fig. 2.8 Segment boundaries naturally follow depth discontinuities.

The disadvantage is that the regions do not contain much feature within themselves. They can be properly analyzed only if their neighbouring regions are substantially different and thus making the boundary areas good features. Obviously, that cannot be guaranteed. For regions with weak features at boundary areas, which are common in over-segmenting, the featureless problem introduced in section 2.3.1 can be significant.

2.6 Review Summary

In this chapter, most popular stereoscopic depth estimation approaches have been covered. Both the global methods and the local methods rely on the optimization of enforcing the uniqueness and continuity constraints. In fact, D. Scharstein and R. Szeliski have shown that the global methods and the local methods are equivalent in certain aspects [SS02]. Our subsequent proposed method is no exception. The multi-stage segment-based architecture handles the continuity with inherent stability, thus resulting in an improvement of performance.

Note that all the methods are limited to their respective context. As with any image processing tasks, different types of input images will most likely require different processing procedures and parameters. Thus, it is a researcher's task to investigate these delicate differences and make his or her method as generic as possible.

Chapter 3

Proposed Multistage Algorithm

Many existing stereo depth estimation algorithms were surveyed in chapter 3. Obviously, there is no trivial solution to the problem. The essence of these methods is to find an optimal balance between two counteractive forces. With that in mind, we shall present our research: a multistage segment-based stereo depth estimation algorithm.

The organization of this chapter is as follows. In section 3.1, the concept of the multistage architecture is introduced. Several basic aspects of image features are discussed in section 3.2. In section 3.3, match cost aggregation methods are examined. The advantage and basic design of segment-based estimation are discussed in sections 3.4 and 3.5, respectively. Section 3.6 is focused on the importance of left-right consistency enforcement. Section 3.7, integrates all these individual pieces together and presents the design in its entirety, which is followed by the chapter summary.

3.1 The Multistage Architecture

Divide and conquer is a strategy often used in signal processing. For example, in audio compression, signals are first divided into several frequency sub-bands and then transformed. This process is simpler and more efficient than attempting to transform the entire band at once.

The same idea can be applied to stereo depth estimation. For area-matching algorithms, a single set of parameters do not perform well for the entire image. As mentioned in chapter 2, other researchers have proposed to use windows of varying sizes and positions to improve performance. That can be viewed as implicit ways of divide and conquer, since different sets of parameters are applied to different parts of the source image.

A more explicit and systematic approach, we assume, should improve the performance further. Hence, we propose to solve the problem with a multistage structure, as shown in Fig. 3.1. In the proposed algorithm, each stage, by itself, uses known area-matching algorithms to estimate the depth map. The real novelty resides in the generic architecture.

All the stages, except the last one, share the same platform and only parameters are differently set. Therefore, the design is simple and the software modules are reusable. The multistage algorithm receives the original image pair as input. The calculated partial disparity maps from each stage are then put together to form the final completed disparity map.

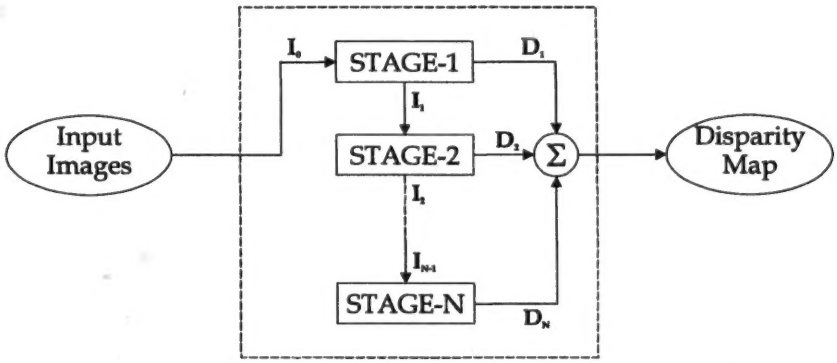


Fig. 3.1 General structure of a multistage approach.

The basic processing procedure of a typical rectangular window matching SDE is depicted in Fig. 3.2 (a); the procedure of any one stage in our multistage structure is shown in Fig. 3.2 (b). Let us consider the first stage for example. It performs area matching on $\mathbf{I_0}$, the input image. Instead of a conventional winner-takes-all decision making for every match attempt, the algorithm allows a no-winner situation. For any pixel or area, if the best matching cost is lower than a certain threshold, the disparity value associated with that cost is accepted. Otherwise, that part of the image is regarded as undecided. Therefore, the first stage has two outputs: $\mathbf{D_1}$ the partially completed disparity map and $\mathbf{I_1}$ the partially processed intensity image. The latter is also referred as the residual image.

Each pixel of the residual image, in addition to its intensity value, is also labelled as being either "decided" or "undecided." Then, $\mathbf{I_1}$ is passed from the first stage to the second stage as the input image of the latter. The second stage, during its own area defining process, will focus on "undecided" pixels only.

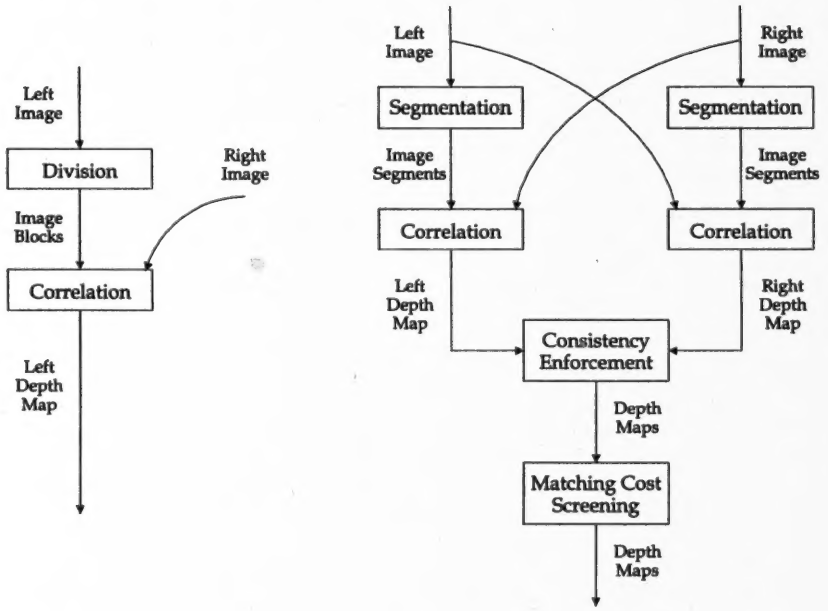


Fig. 3.2 A flow chart of disparity calculation steps: (a) a typical rectangular window based shift-matching algorithm, (b) one stage of the proposed segment based algorithm with left-right consistency enforcement and matching cost screening.

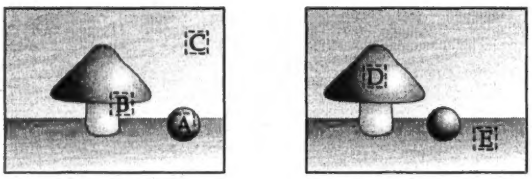
The same process repeats through the stages sequentially. Only basic parameters are changed for each stage. Generally speaking, large areas with uniform intensities are processed in the first two stages and tiny details are processed later—from low frequency signal to high frequency signal, in a broad sense.

The last stage is different from all other stages. Being the last one in the sequence, it cannot leave a residual image. All undecided pixels are to be determined in this stage. Since all the large and easy areas are determined in earlier stages, the input residual image I_{N-1} obviously comprises a large amount high-frequency signals. The matching algorithm employed, thus, needs to be simple and robust.

3.2 Image Segmentation

3.2.1 Image Feature Basics

In the general sense, the term "feature" means "a typical quality or an important part of something" [INT02]. But, in image processing, this term has a specific meaning. A region of an image is well featured if there is significant intensity contrast among its pixels. If the intensity levels are uniform, then that region is featureless. Consider an image pair shown in Fig. 3.3 and let us examine a few important aspects of image feature: size, intensity variation, boundary and direction.



Left Image Right Image

Fig. 3.3 Several aspects of image features.

Size

The more pixels there are in a region, the more likely there is significant contrast. Thus, a large region is more likely to be well featured.

Intensity Variation

In quantitative terms, image feature can be modeled at each pixel, by the magnitude of the gradient function of the image's intensity function $I(x,y)$.

$$|\nabla I(x,y)| = \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2} \qquad (3\text{-}1)$$

For example, in Fig. 3.3, block A possesses significant intensity variation; block C is featureless. Since most SDE algorithms consider horizontal pixel/block matching only, Eq. 3.1 can be simplified to Eq. 3.2. This is a special case, when the vertical partial derivative is not considered.

$$\left|\frac{\partial I(x,y)}{\partial x}\right| \tag{3-2}$$

The intensity contrasts of block E are purely along the vertical axis, and thus it is considered featureless horizontally. Block D, on the other hand, is regarded as a good image feature.

Boundary

In Fig. 3.3, block B has good intensity contrast and thus it is well featured. However, it is trivial that image feature will not help when depth plane boundaries are present; parts of block B will unavoidably be wrongly matched. Therefore, for shift matching purposes, image feature is only useful when the pixels causing intensity contrast is on the same disparity plane. In other words, they are not on a depth boundary.

3.2.2 Segment Correlation Methods

Segment correlation is very similar to rectangular window correlation. Basically, for any given disparity value d, the matching cost of each pixel is summed. There is a minor difference though. The sum of the matching cost must be divided by the segment's size to get the average cost. Otherwise, there is no way we can compare the cost of a random sized segment to a constant threshold.

Matching Cost

Let the matching cost function for a window-matching SDE algorithm can be designated as $C(d)$. By convention, a larger cost implies a low correlation and thus a poor match. A smaller cost, on the other hand, implies a high correlation and a potentially good match.

Usually, d, the disparity variable, is bounded by predefined constants. Let us define $d \in [d_{\min}, d_{\max}]$. This greatly reduces the number of candidates for d and thus prevents the processing time from being unnecessarily long. The optimal disparity is obtained when:

$$C(d_{opt}) = \min\{C(d): d_{\min} \le d \le d_{\max}\} \tag{3-3}$$

Segmentation's Impact on Cost Curves

Although there is always a minimum point for the cost function, there may not be a successful match. A failure can be caused by one or more factors. Four common scenarios of cost function curves are illustrated in Fig. 3.4. An analysis of their cause and their effect should give us more insight.

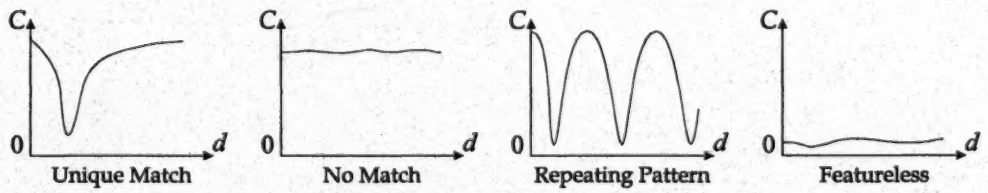


Fig. 3.4 Scenarios of cost curves.

A good match, by definition, should have a low matching cost. Thus, even the minimum point on the function curve is not acceptable, if it is not small in an absolute term. In addition, a good match should also be unique. In other words, the minimum point ought to be the only point with such a small cost.

Let us first examine the four common scenarios of cost curve illustrated in Fig. 3.4. First, a unique match is always desirable. In this case, there is one clear global minimum. The resulting disparity is most likely to be accurate. Repeating patterns and featureless regions, on the other hand, would cause severe aliasing effects that make a correct decision almost impossible. When a winner-take-all algorithm is performed, the final result would be dictated by random factors such as the signal noise.

The "no match" case is fairly common for partially occluded regions or image with a high signal noise level. Although there is no match at all, this situation is actually more desirable than aliasing instances. Knowing there is not a good solution is always preferable to settling on an alias disparity value. The segment-based correlation method has a natural tendency of reducing the possibility of aliasing to "no match" status. More details can be found in the next section.

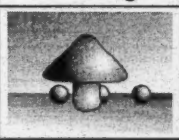
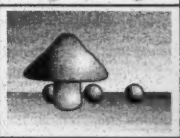
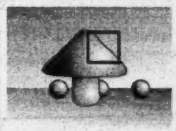
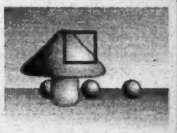
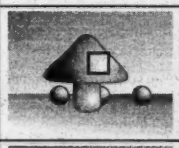
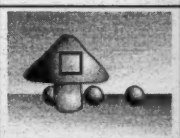
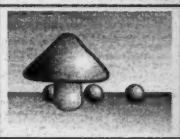
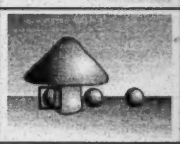
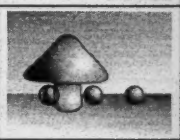
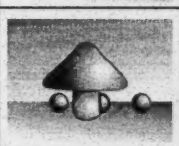
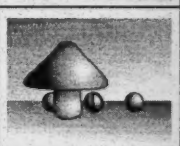
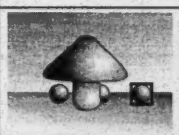
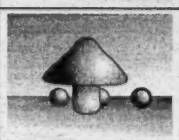
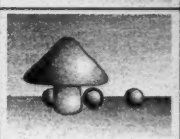
If aliasing is indeed eliminated or at least reduced, we shall concentrate on "no match" situations. Obviously, a simple winner-take-all selection mechanism will not be enough; a more sophisticated approach ought to be used. Since such situations are likely caused by either occlusion or signal noise, it may be theoretically not possible to find a segment's corresponding counterpart. However, it is still possible to assign a correct disparity value for that segment by enforcing the smoothness constraint. Then, the correct disparity values can propagate from "unique match" regions to "no match" regions by aggregation means.

The last diagram in Fig. 3.4 represents the cost curve of a typical featureless rectangular window clipped from a large featureless object. There is a fine line between a featureless window and a featureless object. The former has no FC between itself and its surrounding. The latter has a well-defined one, since a segment tends to reflect the true shape of a natural object. Therefore, the use of segmentation can effectively reduce the occurrence of featureless regions. The large featureless objects may fall into the "unique match" category or, if occlusion and boundary situation are dominating, the "no match" category.

Furthermore, a unique match is not always a correct match. A combination of occlusion and repeating patterns can create a single low matching cost that is actually a wrong match. Table 3.1 describes the common scenarios of feature matching. It is quite obvious that segments have a distinct advantage over rectangular windows, in reducing potential aliasing problems.

Table 3.1

Common Matching Scenarios: Rectangular Windows vs. Segmentation

Case	Left Image	Right Image	Depth Map	Comments
0				The original image pair and the ground truth map of the left image.
1				The window contains two regions of different disparities. A *unique match* is obtained. The entire window is assigned the disparity for the dominate region.
2				The enclosed area is uniform in disparity. However, the window contains little feature. A *featureless* window is likely mismatched to an alias.
3				Segmentation defines the matching area better than rectangular windows. The problems in both case 1 and case 2 are likely to be eliminated.
4				When the selected object is occluded in the right image, a *no match* is likely reported. The resulting disparity value is usually way off the mark.
5				If a segment is used instead of a rectangular window as in case 4, there will be no improvement.
6				When the occlusion happens in the left image, a *no match* situation is again reported.
7				If a segment is used instead of a rectangular window as in case 6, the problem goes away. The disparity in a segment is uniform and thus occlusion is avoided.
8				When a *repeating pattern* is encountered, an alias may be mistakenly selected.
9				The use of segmentation does not eliminate aliasing.

3.2.3 Unique Advantages Of Segments

As mentioned in the previous chapter, most area-matching algorithms simply use windows—rectangular shaped areas. The size and position of a window may vary to improve performance, but certain limitations still apply. Image segmentation, however, can completely avoid some of these problems and reduce some others. Although segmentation is not a new proposition in stereo depth estimation, there is still the need to elaborate on this subject. The reviewed literatures have very little discussion regarding the intrinsic properties of segments. Most algorithms treat tiny segments as basic elements in a global energy function, while we are going to perform area-matching with the segments.

All stages except the last one utilize a segment-correlation method. The entire image is segmented into multiple random-shaped and random-sized regions. Each region, then, undergoes a horizontal shift-match process. The total matching cost of a segment S can be calculated as:

$$C_{segment}(d) = \frac{1}{N} \sum_{pixel \in S} C_{pixel}\left(I_R(x-d, y), I_L(x, y)\right) \tag{3-4}$$

where $C_{segment}$ is the matching cost for the entire segment, C_{pixel} is the matching cost for any pixel, S represents a segment in the left image $I_L(x,y)$, N is the number of pixels in segment S. Since a segment can be of any size, when a threshold is concerned, we have to use a cost-per-pixel measure, instead of a total cost. Thus, the summed cost is divided by the N.

The matching mechanism of the segment-correlation method is similar to that of an ordinary window-correlation method. However, there are some intrinsic differences that shall not be overlooked.

First, a window-correlation method will have to consider situations where multiple disparity regions exist in a single rectangular window. As previously mentioned, there are means to lessen boundary effects, such as adjusting window sizes and positions.

With the segment-correlation method, a segment contour normally agrees with disparity boundaries. The size and position adjustments of a segment come naturally with a properly selected segmentation algorithm. Therefore, homogenous disparity can be realistically assumed within one segment.

Second, and more importantly, a traditional rectangular window is supposed to contain a substantial amount of feature to function properly. However, a segment is expected to contain little to no feature. The real features of a segment-correlation method exits along segment boundaries.

3.3 Implementing Segmentation and Matching Algorithms

Two segmentation algorithms are implemented: region growing and vector quantization. Experimental result has shown that the two methods are very close in performance. The region growing method is chosen to be used in the final setup, because it is easier to implement and faster to run. Since our focus is not about image segmentation, but stereo depth estimation. The segmentation algorithms are not intended to be very sophisticated; they are kept fast, simple and robust.

3.3.1 Region Growing

Since our depth estimation algorithm is intended to be fully automatic, the region growing algorithm must operate with no human input also.

Seed selection

The image is scanned from left to right and from top to bottom in a "Z" shaped path. The first pixel encountered that is not part of an existing region is regarded as a seed.

Growing

The seed itself can be viewed as a region with only one pixel. For any region, all of its neighbouring pixels not yet grouped are investigated. Ones that satisfy the pre-defined threshold are included to the region and labelled as grouped.

Threshold

A pixel can be added, or grown into, an existing adjacent segment if intensity criteria are met. Let T be the scalar threshold value, the relative RGB weight vector be $\overline{w} = [0.299 \quad 0.587 \quad 0.114]$, the intensity of the pixel be $\bar{i} = [r \quad g \quad b]$ and the average intensity of the segment be $\bar{i}' = [r' \quad g' \quad b']$. The pixel can be included into the segment if the following two threshold criteria are met:

$$\max\left(|r - r'|, |g - g'|, |b - b'|\right) < T \tag{3-5}$$

$$\left|\bar{i} - \bar{i}'\right| \cdot \overline{w}^T < 0.8T \tag{3-6}$$

During the implementation and testing, T is set to be from 9 to 13 for different stages.

3.3.2 Vector Quantization

In vector quantization (VQ), the first step is to generate the codebook, which is a suitable set of N colour vectors. In our implementation, we used $N=9$. Initially, the colour vectors in the codebook are set as $\{\phi_i,\phi_i,\phi_i\}$, $0\le i\le 8$ where

$$\phi_r = r\cdot\Delta+\Delta/2, \Delta = 256/(N-1) \tag{3-7}$$

Each pixel (r,g,b) in the image is assigned the colour vector with the minimum Euclidian distance $\left(\sqrt{(r-r')^2,(g-g')^2,(b-b')^2}\right)$, where (r',g',b') is a colour in the codebook. The vector itself is then updated to be the centroid of these assigned pixels. This process is executed iteratively until there is no more change in vector assignment or when a preset number of updates are reached.

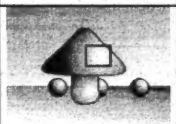
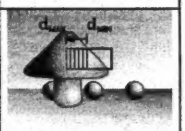
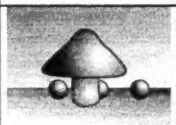
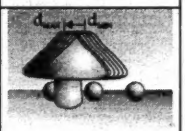
After the codebook is calculated, the entire image is divided into multiple segments. Each segment contains pixels corresponding to only one vector. Any two adjacent segments are to be merged, if both of them correspond to the same colour vector.

3.3.3 Segment Matching

When images segments are obtained, a correlation-based matching process begins. Let us designate the left image as the reference and the segments are results of segmenting the left image. Then, the best match in the right image is to be found. Note that the right image is not segmented. We do not compare a segment from the left image to a segment from the right image. Instead, the segment in the left image is treated as a "window" and that window is applied to the right image to define the pixels to be compared.

Table 3.2

Window Matching vs. Segment Matching

Region Type	Left Image	Right Image
Rectangular Window		
Segment		

The matching cost for the entire segment at a specific disparity d is given by Eq: 3-4. Within the searching range, from d_{min} to d_{max}, the disparity with the smallest cost is returned as the top candidate. The process used to determine the searching range is explained in detail in section 3.7.2.

3.4 Left-Right Consistency Enforcement

Symmetric processing is very useful in local methods. Fig. 3.5 illustrates a stereo pair: the top left and bottom left images represent the right and the left views, respectively. Segments A and B are corresponding counterparts. As shown, we can either shift segment A rightward to match B, or shift segment B leftward to match A. Some window-correlation methods use this symmetric matching strategy to find a disparity map for each image and aggregate the results.

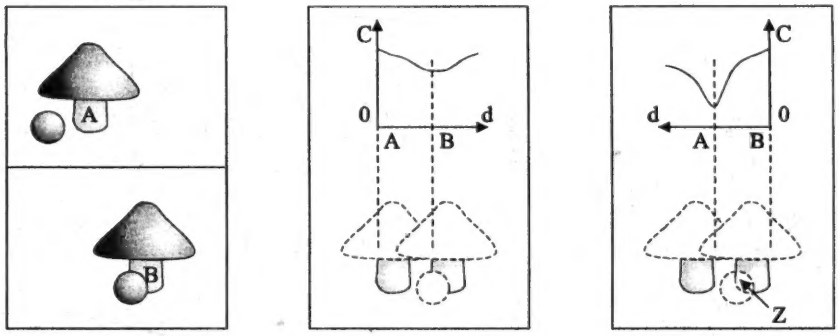


Fig. 3.5 Asymmetric cost curves.

Let us designate $R(x, y)$ and $L(x, y)$ as the disparity maps for the right image and the left image, respectively. They are calculated by shift-matching algorithms discussed previously in section 3.3. Then, we are able to render a second right disparity map $R_L(x,y)$ from $L(x,y)$:

$$R_L(x - L(x, y), y) = L(x, y) \tag{3-8}$$

Similarly, the left-view rendering of the right disparity map is:

$$L_R(x + R(x, y), y) = R(x, y) \tag{3-9}$$

The domain of the new disparity maps is not identical to that of their respective counterparts. Yet, they overlap for most areas. Ideally, where they do overlap, the disparity values should agree exactly. If not, then a selection algorithm must be used to decide which one is better.

Occlusion Handling

When occlusion is present, the cost curve is likely to be the "no match" type. In that case, even the aggregated result is unlikely to be correct. The combination of sym-

metric processing and segmentation can solve some simple cases of occlusion. In Fig. 3.5, while shifting segment A rightward to match B would not yield a unique good match, the reverse process is much better. Being a subset of A, segment B does have its unique perfect match. Instead of assigning an inaccurate disparity for the entire segment A, now the non-occluded sub-region B can be accurate. In short, for that same mushroom root, $R(x,y)$ is undefined for A and $L(x,y)$ is correctly defined for B.

In addition, the occluded region Z is undefined for L and waits for further processing. Since Z is occluded, local methods will not do any good. The only solution is the enforcement of the smoothness constraint. The natural solution would be propagating the disparity value of segment B to Z. Such propagation would be too complex to implement on the basis of image segments. Instead, a much simpler, pixel-based algorithm is used.

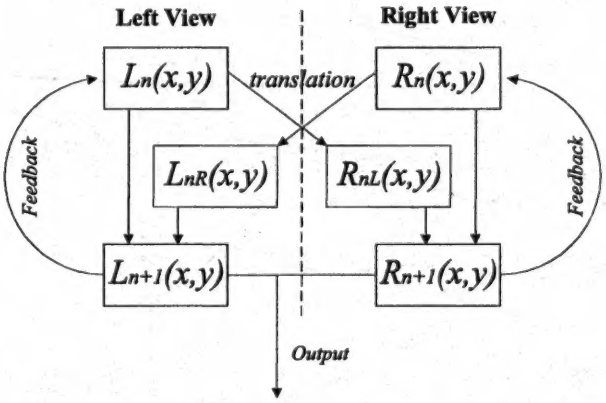


Fig. 3.6 Feedback loop for left-right consistency enforcement.

Define n as an iteration index starting at 1. As shown in Fig. 3.6, both the left and the right disparity maps L_n and R_n are first transformed into the other view, following the rules defined by Eq. (3.3) and Eq. (3.4).

The resulting transformed disparity maps R_{nL} and L_{nR} are compared against R_n and L_n, respectively. The two disparity maps of the same view are then merged into one. When inconsistency occurs, the disparity with the smaller matching cost is selected. Let $C(x,y)$ be the matching cost function. Then the following rules are used during merging:

$$R_{n+1}(x,y) = \begin{cases} R_n(x,y) & \text{if } C_{Rn}(x,y) < C_{RnL}(x,y) \\ R_{nL}(x,y) & \text{otherwise} \end{cases} \quad (3\text{-}10)$$

$$L_{n+1}(x,y) = \begin{cases} L_n(x,y) & \text{if } C_{Ln}(x,y) < C_{LnR}(x,y) \\ L_{nR}(x,y) & \text{otherwise} \end{cases} \quad (3\text{-}11)$$

During implementation, the view translation must be performed from the minimum disparity value, in ascending order, to the maximum disparity value. In the case of n→1 correspondence, the point with a higher disparity is always closer to the camera and thus should be covering one with a lower disparity.

The feedback design is needed, since all the consistency enforcement cannot be done in one pass. A disparity value changed in one pass may themselves induce another change in the next pass. On the other hand, convergence is assumed for two reasons. First, matching costs are always positive. Second, each disparity change will result in either a no-change or a decrease of the total matching cost, where the decrement is always a multiple of a pre-defined constant.

Experiments have shown that, for the test images that we use, at most three passes are need before the depth maps are stabilized and no more changes are required.

3.5 System Integration

There are three main steps in the complete process of our stereo depth estimation: pre-processing, matching and post-processing.

3.5.1 Image Preparation

The original input images may have significant signal noise. It is best that the source images are smoothed to reduce noise and thus yield a more predictable result. A simple edge-preserving Gaussian filter is used. The R, G and B color spaces are smoothed independently to obtain a smoothed color image. Let us take the R component for example. Assuming a horizontal row of pixels has intensities [r_0 r_1 r_2 ... r_N], that row shall be divided into two segments [r_0 ... r_{n-1}] and [r_n ... r_N] if abs(r_n- r_{n-1}) > $r_{threshold}$. This process goes on until the line segments can no longer be divided. Each line segment is then Gaussian smoothed by:

$$\begin{bmatrix} r_n' \textit{ left bound} \\ r_n' \textit{ in between} \\ r_n' \textit{ right bound} \end{bmatrix} = \begin{bmatrix} 1/4 & 3/4 & 0 \\ 1/4 & 1/2 & 1/4 \\ 0 & 3/4 & 1/4 \end{bmatrix} \begin{bmatrix} r_{n-1} \\ r_n \\ r_{n+1} \end{bmatrix} \tag{3-12}$$

3.5.2 Initial Disparity Range Estimation

The test images we use typically are around 400 pixels wide. It is unrealistic to shift-match an image segment from 0 to 400. That would be both time inefficient and alias-prone. Therefore, we need to have an educated guess about the range of disparity values for the output depth map. As shown in Table 3.2, a successful matching process depends heavily on the proper estimation of d_{min} and d_{max}. Not only process-

ing time can be saved, obviously wrong matching can be avoided also. An initial guess of d_{min}=0 and d_{max}=$X/10$ is used, where X is the image width in pixels.

A simple window matching algorithm, just like the one used in the last matching stage (see next section), is used. A depth map D and its corresponding matching cost map C are generated. The average cost C_{ave} is calculated. All pixels with a cost less than $2c_{ave}$ are regarded as confident. A histogram function $H(d)$, with d ranging from 0 to $X/10$, is then built on the depth value of these confident pixels. Here, let the threshold h_T be 0.5% of the total number of pixels in the image. The range is then narrowed down from $[0, X/10]$ to $[d_m, d_n]$, if $m<n$, $H(d_m)>h_T$, $H(d_n)>h_T$, $\max(H(0), H(1), \ldots, H(d_{m-1}))< h_T$ and $\max(H(d_{n+1}), H(d_{n+2}), \ldots, H(X/10))<h_T$. The resulting estimated range is quite narrow and precise; more testing details will be revealed in chapter 4.

3.5.3 Segmentation and Matching

The first and second stages use region growing and vector quantization segmentation respectively. Region matching is then performed on the obtained segments. However, with the proposed method, these two stages will likely leave a partially finished image; all the residual information is then dealt in the last stage. Especially when the image quality is substantially poorer than expected, most pixels of the initial image will be marked as "low confidence." Thus, in the worst case, the last stage has to be ready to accept almost the entire initial image and leave no residue image after processing.

Since all the previous stages have failed to process these residue pixels, other types of complicated algorithms are not likely to work either. A simple, yet robust, method is needed. Hence, we choose the very basic window-correlation approach,

with an absolute difference cost function and a WTA selection mechanism. This method is very basic. While it may not yield great results for good quality images, it definitely gives reasonable results for all images.

The last stage should not leave a residue image pair behind; all undecided regions should be calculated here. As mentioned before, this has been a new trend in the depth analysis community as computer technology advances. Yet, this area is still new enough to have great potential. We will focus on natural images with reasonable amount of texture and occlusion. Since the scope of research is determined, we can make reasonable assumptions to simplify the problem.

To preserve the natural edges better, circular windows are used. Experimental results have shown that, although the error rate of circular windows is not significantly better than that of rectangular windows, the visual effect is indeed much better. The diameter of the window is set to be 5 pixels, while the searching range is still from d_{min} to d_{max}.

In order to eliminate irregularities at object boundaries, the resulting completed depth map undergoes a smoothing algorithm. Again, circular windows with a diameter of 5 are used. All the pixels within the windows are assigned the cost and disparity value of the pixel with the least matching cost.

3.5.4 Post-Processing

Artefact Removal

When a segment mismatch occurs due to occlusion or aliasing, there will likely be a significant error in its estimated disparity. Visually, a certain part of the depth map shows a sharp contrast of intensity levels.

Depth estimation error can happen to segments of any size and the degree of error can vary. However, the term "artefact" typically refers to smaller segments with huge errors, as illustrated by Fig. 3.7. The artefact removal algorithms, thus, is based on such understandings.

Ground Truth DM Estimated DM

Fig. 3.7 Examples of artefacts.

There are three steps involved to remove artefacts: segmentation, size evaluation and artefact recovery. The first step is straight forward. The depth map is divided into multiple segments. Two pixels are considered to be in one segment only if they are adjacent and they have identical disparity values. This process should be rather straight forward.

The second step is to compare the size of the segment, in terms of number of pixels, against a pre-selected threshold T. If the segment is over the threshold, it is considered not an artefact. If it is correctly evaluated, no operation is needed. Otherwise, it still cannot be safely removed due to its large size. The third step removes the previously identified artefacts from the depth map. A simple line-fitting is used to fill in the newly generated gaps. For each row of pixels in this empty region, the intensity values transients linearly from left to right.

Line Segment Smoothing Algorithm

To reflect the subtle depth transitions in a relatively smooth region, a line segment smoothing algorithm is used. Again, the assumption is that smoothness in intensity should correspond to smoothness in depth also. Each horizontal scan line is divided into several shorter segments, such that, for any two adjacent pixels in one line segment:

$$|r_n - r_{n-1}| + |g_n - g_{n-1}| + |b_n - b_{n-1}| < T_{line} \tag{3-13}$$

where T_{line} is an intensity threshold. In our implementation, it is set at 14. Once the lines are broken down properly, the disparity value for each pixel within a segment is gathered to make a histogram. Then, d_{max}, the disparity value with the most entries is regarded as the predominate value. The entire line segment is then set to d_{max}.

3.6 Summary

In this chapter, we have introduced the concept of a multistage stereo depth estimation algorithm. After investigating multiple mismatch and occlusion scenarios, the theoretical advantages of multiple stages over a single stage are explained in detail. The image segmentation process and the segment matching algorithm are the main body of our proposed method. The former uses very simple region growing and vector quantization algorithms, while the latter is similar to that of a conventional rectangular window's matching algorithm.

This chapter is concluded by a thorough description of the pre-processing step, which involves an improved Gaussian filtering, and the post-processing step, which involves segment-based artefact removal and line-based low-pass filtering.

Chapter 4

Performance Evaluation

As we have proposed a novel multistage approach for stereoscopic depth estimation in the previous chapter, a thorough evaluation is needed to justify the use of this new algorithm. In this chapter, we will first introduce the testing method and the performance metrics we are going to use. The performance of the proposed method is then presented in terms of these metrics. Comparisons with other techniques of a similar genre are made to reveal the improvements. At last, qualitative merits of the proposed algorithm are examined by a visual evaluation of the calculated depth maps.

4.1 Testbed Specifications

Historically, stereoscopic depth estimation had lacked a standard numerical evaluation method. Early works used visual evaluations only. In 2002, D. Scharstein and R. Szeliski [SS02] established a quantitative software testbed for SDE. Many researchers have hence adopted the evaluation mechanism and submitted their results to the

Middlebury Stereo Vision Page. In this paper, we are going to use the same test images, test procedure and performance metrics, so that our results can be compared with existing algorithms directly. Our test program is implemented in Java. It has been thoroughly tested that its output is identical to the Middlebury results.

4.1.1 Test Images

There are four standard stereoscopic image pairs included in the test bed. These are the Tsukuba, Saw, Venus and Map image pairs from the Middlebury website [INT01].

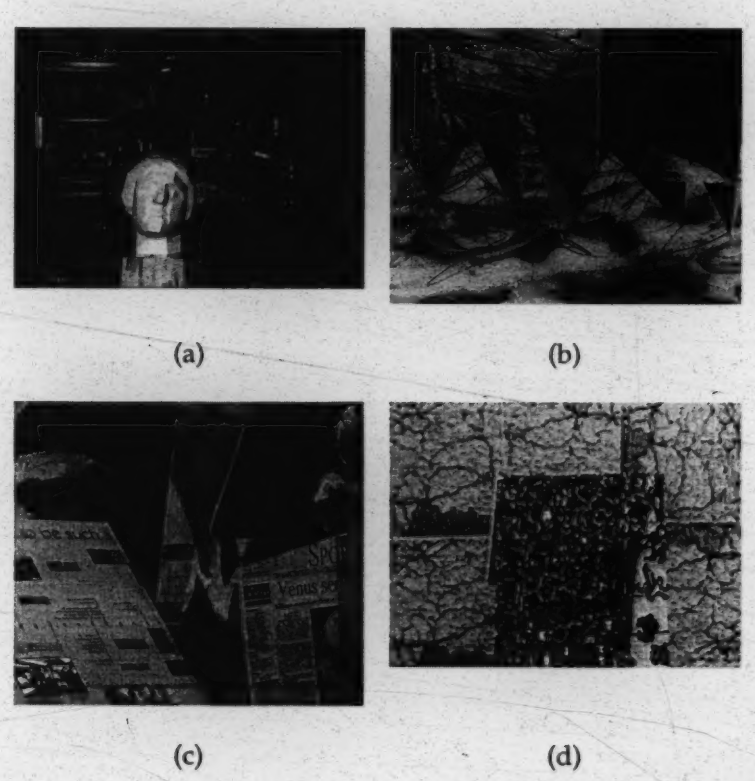


Figure 4.1 Left images of the test image pairs: (a) Tsukuba, (b) Sawtooth, (c) Venus, (d) Map

All the test images are calibrated left-right pairs. In other words, only horizontal disparities are considered; vertical disparities are made to be zero at all times. These are all color images, except that the Map pair is grey level.

The images are shown in Fig. 4.1. The Tsukuba depicts a typical laboratory room. It is very complex in structure, containing many natural objects of different depth. Both the Saw and Venus image pairs are composed of simple planar surfaces. However, the surfaces are complex in texture region. Map, the last image pair, is grey level. It depicts one road map on top of another. The image pair has very cluttered textures. There are substantial signal noises and occlusion areas present.

4.1.2 Disparity Value Ranges

The maximum and minimum disparity values are known for each test image. For example, as shown in Table 4.1, the disparity for Tsukuba is between 5 and 14 pixels. However, our algorithm should not take that information directly. The disparity range is estimated by an algorithm stated in section 3.7.2.

Table 4.1

Disparity Ranges for Test Images

	Minimum Disparity	Maximum Disparity
Tsukuba	5	14
Sawtooth	4	18
Venus	3	19
Map	4	28

4.1.3 Performance Metrics

In this paper, we use the *percentage of wrongly matching pixels* (PWMP) as the performance criterion. It is a standard established by the Middlebury test bed [INT01]. For a stereo image pair, we estimate the disparity for each pixel in units of pixels. Let the estimated disparity, and the true disparity of pixel (m,n) be denoted by $d_e(m,n)$ and $d_t(m,n)$, respectively. For an image with size $M \times N$, the PWMP is defined as:

$$\eta = \frac{100}{M \cdot N} \sum_{(m,n)} \left(\left| d_e(m,n) - d_t(m,n) \right| > \delta_d \right) \tag{4-1}$$

where $1 \le m \le M$, $1 \le n \le N$, and δ_d is a threshold for error tolerance. In this thesis, we set δ_d as 1.

Note that the performance of an algorithm is sensitive to many image parameters such as texture, and edges. Occluded pixels are never considered in calculation PWMP, as specified by the Middlebury test bed [INT01]. For a given stereo image pair, we calculate η for three types of regions.

- Entire Image, known as "ALL". This provides the overall performance.
- Textureless regions, known as "TL". Here, the horizontal gradient is small.
- Depth discontinuity regions, known as "DISC". Here, the neighboring disparities differ substantially.

4.1.4 Test Procedures

The definition of featureless and discontinuity regions are explained by [SS02]. For all practical purposes, we only need to use the mask images they have provided.

Fig. 4.2 shows mask images for the four test image pairs: Tsukuba, Sawtooth, Venus and Map. There are three mask images for each of the test image pairs. Each

mask images can only take on three intensity levels 0 (black), 127 (grey) and 255 (white). The black pixels belong to either boundaries or occluded areas; they are ignored when calculating PWMP. The grey pixels belong to non-occluded areas that do not qualify. For example, non-boundary pixels do not qualify in the DISC mask. They are ignored when calculating PWMP, too. Only the white pixels are qualified and thus considered when calculating PWMP.

4.2 Numerical Results

Numerical measure is needed to evaluate our proposed method. By quantifying the results, we can then know the strength and weakness of the multistage architecture and how it compares to existing algorithms.

4.2.1 Advantages of Multistage

Table 4.2 reveals the performance advantage of the multistage method. No post processing is used, in order to magnify the gap of performance. Since stage 3 is the last stage, it is intended to process all residue information. Thus, stage 3 is always used. By itself, stage 3 is a typical single stage window matching algorithm. We can see a clear pattern: with the help of the segmentation stages, the percentage errors decrease substantially.

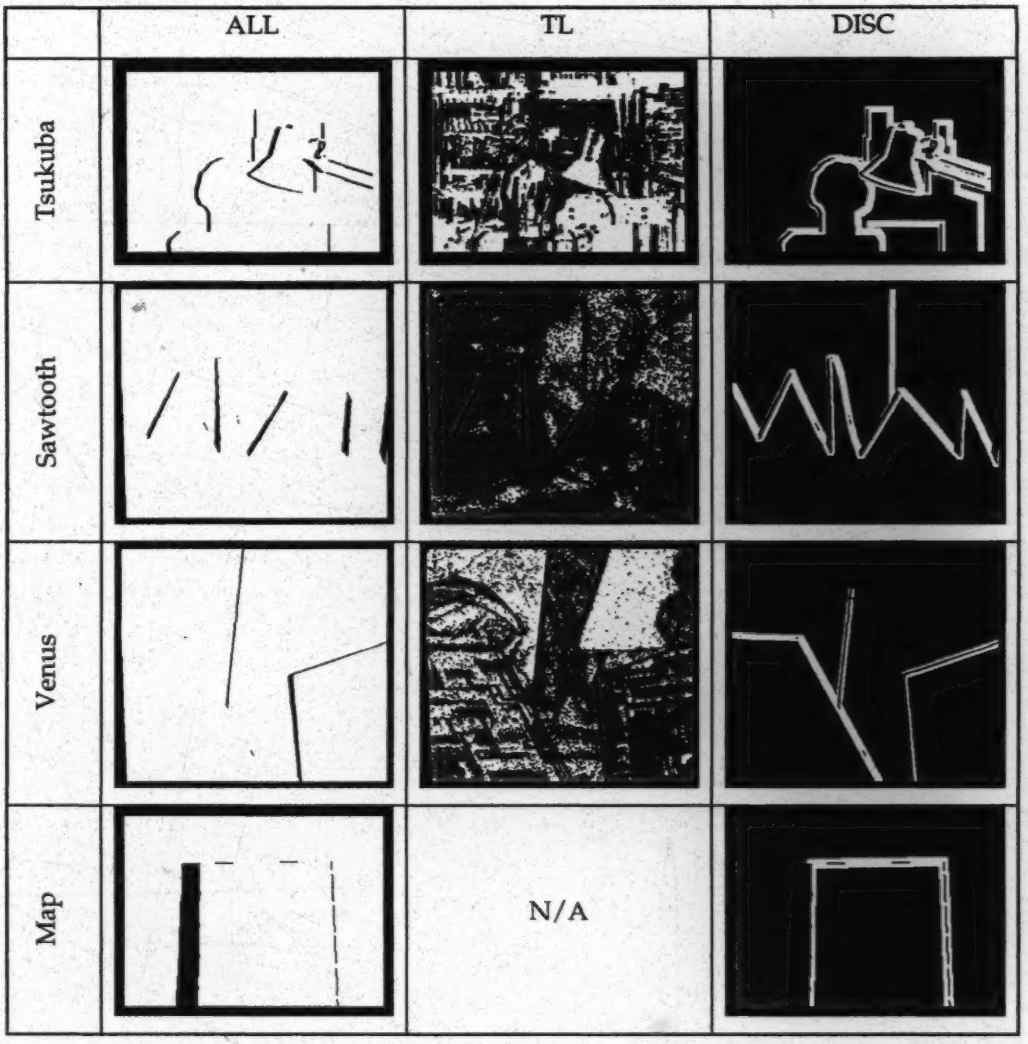


Fig. 4.2 Mask images for evaluation purposes.

Table 4.2

Performance (η) Improvements Using Multiple Stages

	Error Rate		
Stage 1	×	√	√
Stage 2	×	×	√
Stage 3	√	√	√
ALL	9.26	5.40	4.94
TL	16.9	8.29	6.69
DISC	13.3	12.0	12.9

Since we allow a partially finished disparity map from each stage (except the last one), it is important to know exactly what percentage of the test images are determined. Table 4.3 shows the percentage of the completion for each stage and its corresponding error rate. For an entry P@Q, P corresponds to the PWMP of the completed area; Q corresponds to the size of the completed area as a percentage of the entire image. The estimation efficiency is calculated before applying post processing. It is obvious that early segmentation stages are very well suited to un-textured regions with a very low error rate.

Table 4.3

Performance (η) and area of determined pixels (%) for Tsukuba

Mask Image	Stage 1	Stage 2	Stage 3
ALL	0.20 @ 44.6	0.21 @50.2	4.93 @ 100
TL	0.15 @ 70.1	0.13 @ 72.9	6.69 @ 100
DISC	1.61 @ 26.9	1.69 @ 35.3	12.9 @ 100

Table 4.4 includes the results for all four test image pairs. For simplify, only the "ALL" results are shown. It can be observed that if the first two stages can successfully calculate a large percentage of pixels, the final result is likely to be better. Post processing is also very important; it can smooth out spurious pixels and thus greatly improving the overall performance.

Table 4.4

Performance (η) and area of determined pixels (%) for all test images

Test Image	Stage 1	Stage 2	Stage 3	Post Processing
Tsukuba	0.20 @ 44.6	0.21 @ 50.2	4.93 @ 100	1.13 @ 100
Sawtooth	0.19 @ 38.5	0.22 @ 48.7	4.75 @ 100	1.12 @ 100
Venus	0.25 @ 27.9	0.25 @ 29.1	8.17 @ 100	2.83 @ 100
Map	N/A @ 0	N/A @ 0	2.64 @ 100	0.70 @ 100

4.2.2 Comparison with Other Methods

Table 4.5 compares the performance of the proposed algorithm with three other window-based methods: Real-Time Correlation (RTC) based method, fast variable window (FVW) method, and Windows-based discontinuity preserving (WDP) method. The performance of the RTC method was obtained from [HIG02], FVW was obtained from [V03] and WDP performance was obtained from [AD04]. They are among the best algorithms listed on the Middlebury website that are focused on window matching and do not minimize any energy function.

We can see that our proposed method performs much better than the other three methods for the Tsukuba test image pair, where each piecewise disparity surface is relatively uniform in intensity. However, when the image is very cluttered, image processing no long works well and thus causing a relatively poor perform-

ance. For example, the segmentation stages cannot confidently calculate the disparity value for any pixel in the Map pair. As shown in Table 4.3, the completion rates are zeroes. In other word, the result we obtain is purely from stage 3, a typical single stage. Since we used position varying circular windows in the third stage, the results should theoretically be similar to that of the compared methods. The significant gap could be attributed to the post-processing procedures. Our post-processing is quite primitive and has a lot of room for improvement.

Table 4.5

Performance (η) Comparison with Existing Techniques

Image	Pixels	FVW [V03]	WDP [AD04]	RTC [HIG02]	Proposed Method
Tsukuba	ALL	2.35	1.78	4.25	1.14
	TL	1.65	1.22	4.47	0.30
	DISC	12.17	9.71	15	6.25
Saw	ALL	1.28	1.17	1.32	1.12
	TL	0.23	0.08	0.35	0.12
	DISC	7.09	5.55	9.21	6.80
Venus	ALL	1.23	1.61	1.53	2.83
	TL	1.16	2.25	1.80	5.28
	DISC	13.35	9.06	12.33	3.51
Map	ALL	0.24	0.32	0.81	0.70
	TL	NA	NA	NA	NA
	DISC	2.9	3.33	11.4	8.39

4.3 Visual Evaluation

We must not forget that the quantitative evaluation is performed on a simple model. Although it is one of the best and most commonly used, the model is not conclusive. The top ranking algorithms are so close in performance that even a small change in the model definition will result in a large variation in terms of ranking. Our results demonstrate outstanding handling of edges, especially those of rods. The depth maps generated by different stages are shown in Fig. 4.3. These depth maps are in grayscale, which is an integer multiple (16) of the calculated disparity values.

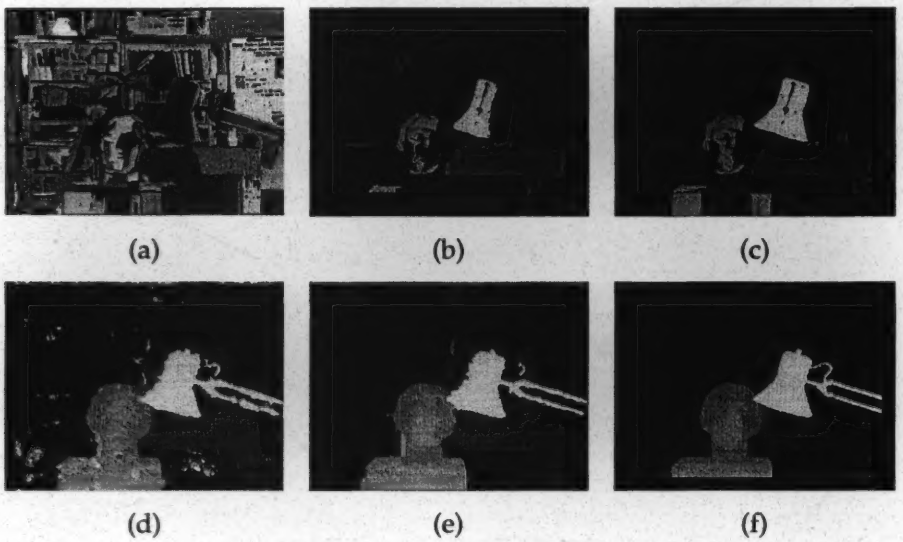


Fig. 4.3 "Tsukuba" output depth maps at different stages: (a) the segmentation result in stage 1, (b) depth map after stage 1, (c) depth map after stage 2, (d) depth map after stage 3, (e) final depth map, (f) ground truth depth map.

Four "Tsukuba" depth maps generated by different SDE techniques are shown in Fig. 4.4. Although their PWMP values only differ slightly, the visual qualities are quite different. Our proposed method is the only one that properly maintains the shape of the lamp's two arms. We have preserved object edges much better than the other methods. As a result, the entire depth map looks clean and resembles the ground truth map the most.

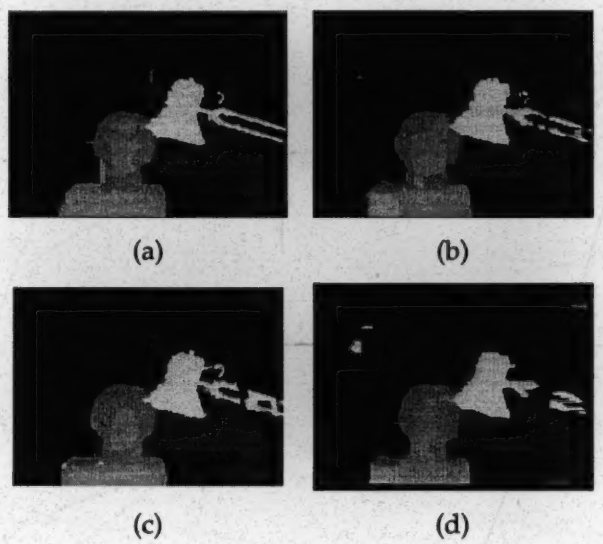


Fig. 4.4 "Tsukuba" depth maps generated by different SDE techniques: (a) the proposed method, (b) FVW [V03], (c) WDP [AD04], (d) RTC [HIG02].

The generated depth maps for Sawtooth are shown in Fig. 4.5. Same as with Tsukuba, the first two segment matching stages have correctly calculated most of the uniform areas. The final result is smooth looking. Intensity edges are preserved very well.

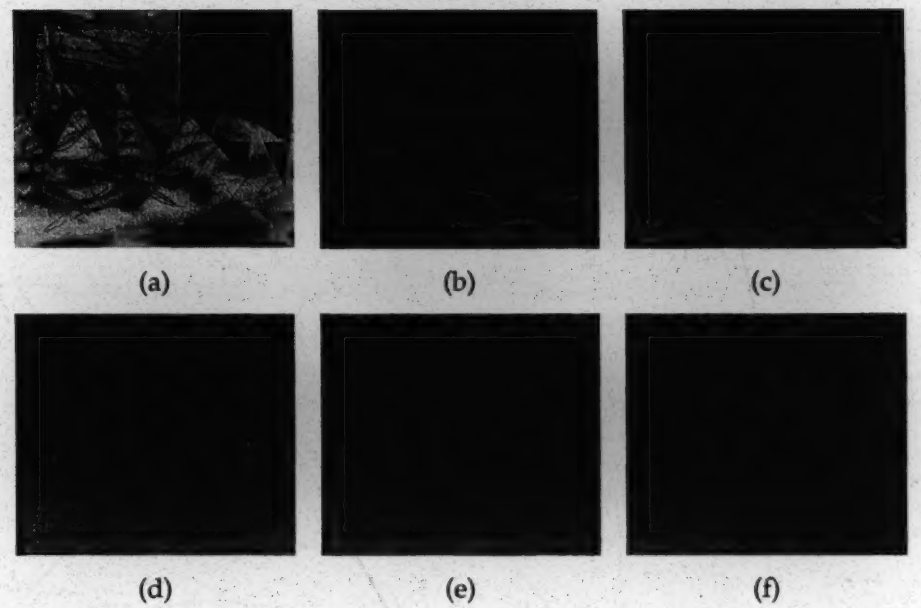


Fig. 4.5 "Sawtooth" output depth maps at different stages: (a) original image, (b) depth map after stage 1, (c) depth map after stage 2, (d) depth map after stage 3, (e) final depth map, (f) ground truth depth map.

Our PWMP performance for Venus is not as good as the listed competition methods, as stated in Table 4.5. From the depth maps shown in Fig. 4.6, the visual quality is reasonably good. The black rectangular artefact at the bottom-left corner is within occluded area. It was not included in the PWMP computation for ALL.

The relatively large error is mainly attributed to the segmentation process. When the segments are too large, there are more than one disparity levels within the segment, even though the transition is smooth. Therefore, assigning the entire segment one disparity value will cause error. This problem can be addressed by using a surface fitting algorithm. More details are given in section 5.3.2.

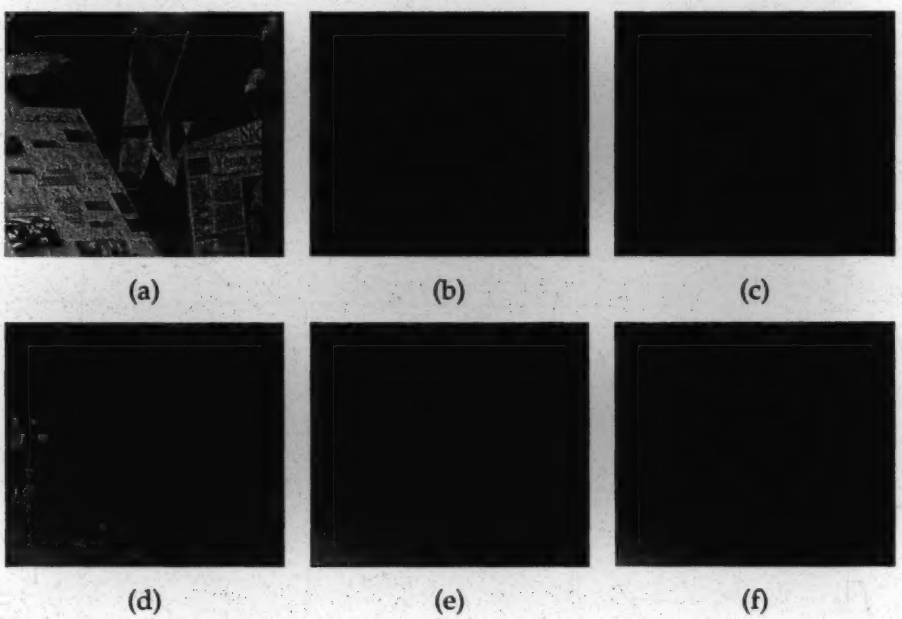


Fig. 4.6 "Venus" output depth maps at different stages: (a) original image, (b) depth map after stage 1, (c) depth map after stage 2, (d) depth map after stage 3, (e) final depth map, (f) ground truth depth map.

As shown in Fig. 4.7, the segmentation stages did no useful processing to Map images. The images are too cluttered to be segmented reliably. The final depth map is essentially the result of the third stage and the post-processing. Again, the depth map looks reasonably good. A surface fitting algorithm can further improve its sub-one percent PWMP performance.

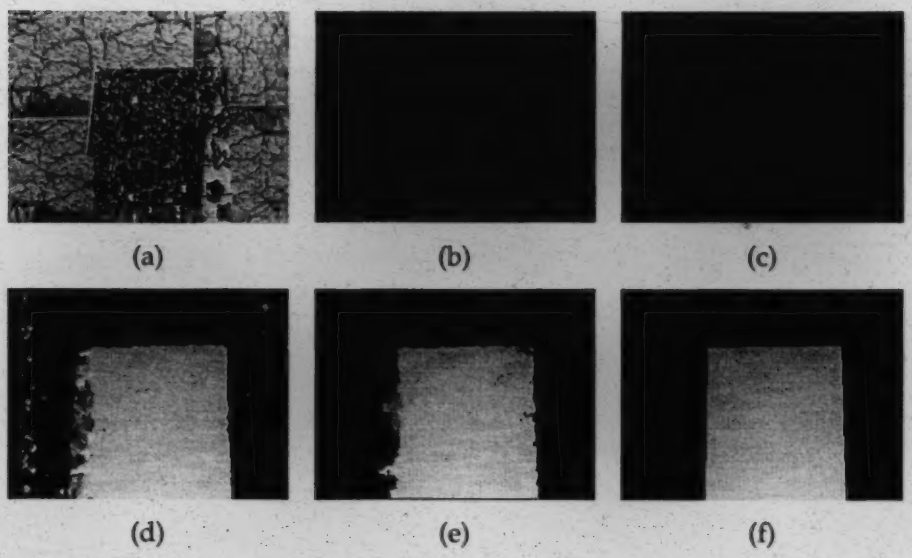


Fig. 4.7 "Map" output depth maps at different stages: (a) original image, (b) depth map after stage 1, (c) depth map after stage 2, (d) depth map after stage 3, (e) final depth map, (f) ground truth depth map.

4.4 Program Running Speed

The testing environment is an IBM PC running Microsoft Window XP Standard Edition. The CPU is an Intel 2.4GHz Pentium 4; memory size is 512M. All source codes are written in Java, and no run speed optimization is done. The compiler and run time environment are both from the JDK 1.4.1 package. Table 4.6 lists the time need to finish each step, for the Tsukuba and Sawtooth image pairs. It is obvious that the

VQ segmentation step used the most time. We can improve the run time performance substantially, if we can use a faster segmentation algorithm.

Table 4.6

Program Running Speed: Time Spent (s) for Each Step

Time Spent (Seconds)	Tsukuba 384 by 288	Sawtooth 434 by 380
Initialization	10	17
RG Segmentation	14	38
VQ Segmentation	71	250
Disparity Calculation (all stages combined)	29	45
Consistency Enforcement (all stages combined)	3	7
Post-processing	17	40
Total	**144**	**397**

4.5 Summary

In this section, we have demonstrated the numerical and visual results of the proposed method. In terms of PWMP performance, there is sufficient evidence that a multistage architecture is superior to a single stage approach. The visual results further demonstrated the advantage of using image segments. Disparity boundaries are well very preserved. We have also noticed that the result is not very satisfying when the segmentation stages do not function well. This problem can be addressed by improving the image segmentation algorithms and the post-processing algorithms. More will be discussed in section 5.3.

Chapter 5

Conclusions

The major goal of this thesis work was to investigate the application of a multistage, segment based architecture in stereoscopic depth estimation. The major contributions of the research work are summarized in section 5.1. Future research directions are identified in section 5.2. The publications resulting from this research work are listed in 5.3.

5.1 Contributions

Stereoscopic depth estimation is a very versatile ranging technique. It is used in many real world applications. In this thesis, we have proposed a multistage, segment based architecture for SDE analysis.

The multistage architecture allows the input images to be processed in several passes. Starting from large and uniform segments, i.e. low frequency signals, our algorithm first identifies regions of high confidence. Then, smaller segments and pixel level image features are processed with different sets of parameters and thresholds.

Compared with traditional single stage methods, our proposed multistage method better suits the characteristics of each sub-region of the input images. The numerical performance evaluation shown in Table 4.2 is a strong proof of the advantages of multiple stages.

The use of segments, instead of rectangular windows, mitigates problems caused by image occlusions and featureless aliasing. The detailed analysis of the transitions among the four common matching scenarios, we believe, is the most comprehensive among recent publications. Table 3.1 is an in-depth account of how segments can avoid or reduce problems, where rectangular windows have failed. Since it is hard to quantify the performance improvement brought exclusively by the used of segments, we have to resort to visual evaluations. As shown in Fig. 4.4, the proposed method outperforms the competition at the edges, especially at the lamp's arm region.

The left-right consistency enforcement module significantly improves the accuracy in processing boundary and occluded regions. During the horizontal shift-matching process, if we shift from a background region (small disparity value) to a foreground region (large disparity value), problems are likely to occur. The disparity assignment for pixels close to the boundary will probably be that of the foreground region, thus creating a large error for the background pixels. The consistency enforcement makes sure that there is one shift-matching from left to right, and another one from right to left. The directional advantage will likely make the better match have a lower matching cost, thus improving the overall accuracy.

5.2 Publications

Part of the work presented in this thesis has been published in the following journals and conferences:

- L. Jia, M. Mandal, and T. Sikora, "Efficient Disparity Estimation using Region based Segmentation and Multistage Feedback," *WSEAS Transactions on Communications*, Vol. 5, Issue 9, pp. 1577-1584, September 2006.

- L. Jia, M. Mandal, and T. Sikora, "Efficient Disparity Estimation using Region based Segmentation and Multistage Feedback," *Proc. of the 10th WSEAS International Conference on Communications*, pp. 582-589, Vouliagmeni, Athens, Greece, July 13-15, 2006.

- L. Jia and M. Mandal, "Novel multistage feedback algorithm for stereo correspondence," *Proc. of the IEEE International Midwest Symposium on Circuits and Systems*, Cincinnati, Ohio, USA, Aug 7-10, 2005.

5.3 Future Work

No SDE algorithm is perfect, and there is always room for improvement. Our research is no exception. For example, our segment based aggregation method is supposed to be better than a rectangular window in terms of reducing aliases caused by large featureless regions. However, in some cases, the segment has grown so big that the entire segment can never have a low cost match, if there are some occlusions. The

more we investigate and understand about stereoscopic depth estimation, the more problems surface. These problems are usually very subtle and are correlated with each other. In this section, three possible improvements are proposed. We hope that we can gain more insight into this subject, once the improvements are implemented.

5.3.1 Intelligent Segmentation

The segmentation algorithms used in our research are very basic, for simplicity. More sophisticated algorithms can be developed specifically for our SDE analysis. Here are some ideas:

- The segment size should be managed more intelligently. Extra large segments can be divided into small pieces, while ensuring each piece contains a reasonable amount of image features.
- The segmentation algorithm should have a feedback system. The output depth maps are to be evaluated after each stage. If certain types of matching error are predominate, then the segmentation parameters can be adjusted automatically and then yield better segmentation results.
- Repeating patterns are always difficult to segment, especially when they are small and highly featured. There is the possibility to use fractal analysis to reduce the patterns to uniform regions. This improvement could be significant. But it is also very complex, and represents a large amount of research by itself.

5.3.2 Surface Modeling

Compared with similar local methods, our proposed SDE algorithm performs better for the "Tsukuba" images, where the depth boundaries closely follow the intensity boundaries. However, our method is at a disadvantage, when the image is highly textured with a very simple ground truth depth map. Our refinement strategy, i.e. piecewise linear low pass filtering, does not work well with such situations. A possible solution is to incorporate surface fitting to the refinement process. In that way, not only spurious error spots can be eliminated, but also we can obtain a smoother depth map, resulting in drastically improved sub-pixel accuracy.

5.3.3 Parameter Selection

Due to the limited scope of our research, most of our algorithm parameters are obtained by trail and error methods. In order to be more versatile, a training module can be implemented to make our algorithm choose the appropriate parameters automatically. The module should consider image characteristics such as noise level, color distribution and image size. Then, multiple (far greater than the existing four) pairs of test images are to be used to train the module to generate better results.

Bibliography

[AD04] M. Agrawal and L. Davis, "Window-based discontinuity preserving stereo," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. 66-73, 2004.

[BI99] Aaron F. Bobick and Stephen S. Intille, "Large occlusion stereo," International Journal on Computer Vision (IJCV), 33(3):181–200, 1999.

[BT98] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," ICCV 1998.

[C86] John Canny, "A computational approach to edge detection," Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[CHMR92] Ingemar J. Cox, Sunits Hingorani, Bruce M. Maggs, and Satish B. Rao, "Stereo without disparity gradient smoothing: a Bayesian sensor fusion solution," D. Hogg and R. Boyle, editors, Proceedings of British Machine Vision Conference, pages 337–346, Leeds, UK, September 1992.

[CORG93] P. C. Cosman, K. L. Oehler, E. A. Riskin and R. M. Gray, "Using Vector quantization for Image Processing" Proc. of the IEEE, Vol. 81, pp. 1326-1341, Sept 1993.

[DA89] Umesh R. Dhond and J. K. Aggarwal, "Structure from stereo—a review," IEEE Transactions on Systems, Man, and Cybernetics, 19(6):1489–1510, 1989.

[FRT00] A. Fusiello, V. Roberto, and E. Trucco. "Symmetric stereo with multiple windowing," International Journal of Pattern Recognition and Artificial Intelligence, 14(8):1053-1066, December 2000.

[GLY95] Davi Geiger, Bruce Ladendorf, and Alan Yuille, "Occlusions and binocular stereo," Interna-tional Journal on Computer Vision, 14:211–226, 1995

[GW02] Gonzales and Woods, Digital Image Processing, Prentice Hall, 2002.

[HIG02] H. Hirschmuller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," International Journal of Computer Vision, Vol. 47, No. 1/2/3, pp. 229-246, 2002.

[HS88] Chris Harris and Mike Stephens, "A combined corner and edge detector," M. M. Matthews, editor, Proceedings of the 4th ALVEY vision conference, pages 147–151, University of Man-chester, England, September 1988.

[INT01] http://bj.middlebury.edu/~schar/stereo/web/results.php

[INT02] http://dictionary.cambridge.org/

[KO94] Takeo Kanade and Masatoshi Okutomi, "A stereo matching algorithm with an adaptive win-dow: Theory and experiment," IEEE Transactions on Pattern Analysis and Machine Intelli-gence, 16(9):920–932, September 1994.

[KZ01] Vladimir Kolmogorov and Ramin Zabih, "Computing visual correspondence with occlusions using graph cuts," Proceedings of the 8th International Conference on Computer Vision, Van-couver, Canada, July 2001.

[MP79] D. Marr and T. Poggio, "A theory of human stereo vision," Proc. Royal Society of London, Vol. B204, pp. 301-328, 1979.

[R99] S´ebastien Roy, "Stereo without epipolar lines: A maximum-flow formulation," International Journal of Computer Vision, 34(2/3):147–161, 1999.

[S02] C. Sun, "Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques," International Journal of Computer Vision, vol. 47,

No.1/2/3, pp.99-117, May 2002.

[SS02] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International Journal of Computer Vision, Vol. 47, No. 1, pp. 7-42, 2002.

[TSK01] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar, "A global matching framework for stereo computation," Proceedings 8th International Conference on Computer Vision, volume 1, pages 532–539, Vancouver, Canada, July 2001.

[V03] O. Veksler, "Fast variable window for stereo correspondence using integral images," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. 556-561, 2003.